REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
31 Dec 2004

3. REPORT TYPE AND DATES COVERED
Final Progress Report
10/01/2002
09/30/2004

4. TITLE AND SUBTITLE
Factored Sampling Tracking

5. FUNDING NUMBERS
DAAD19-02-1-0357

6. AUTHOR(S)
A. Kamalvand
P. MacDonald
T. Tran

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Huston-Tillotson College
900 Chicon St.
Austin, Texas 78702-2795

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

44273.1-CI-H

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

In this report, we compare the performance of two tracking technologies, the Kalman filter and a factored sampling procedure known as the Condensation Algorithm, to track a missile in a dense countermeasures engagement. During a missile engagement, countermeasures may produce features that are identical to those of the target. Consequently, correct identification of the target missile from decoys can result in confusion, delays, and lost track. Therefore, the two algorithms were compared on the following bases: (1) the accuracy of identification, (2) the time required to identify the target correctly, and (3) sensitivity with respect to missile and decoy masses and process noise.

14. SUBJECT TERMS

15. NUMBER OF PAGES
28

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	**UNCLASSIFIED**	**UNCLASSIFIED**	**UL**

NSN 7540-01-280-5500

Standard Form 298 (Rev.2-89)
Prescribed by ANSI Std. 239-18
298-102

Technical Report

Factored Sampling Tracking:
Comparison of the Kalman and the Condensation Algorithms for Missile Tracking in a Dense Target Environment

Ahmad Kamalvand
Paul MacDonald
Thai-Duong Tran

December 31 2004

Huston-Tillotson College
Austin, Texas

Approved for public release; distribution unlimited.

Huston-Tillotson College

Factored Sampling Tracking:

Comparison of the Kalman and the Condensation Algorithms for Missile Tracking in a Dense Target Environment

Ahmad Kamalvand
Paul MacDonald
Thai-Duong Tran

Technical Report

31 December 2004

AUSTIN TEXAS

ABSTRACT

In this report, we compare the performance of two tracking technologies, the Kalman filter and a factored sampling procedure known as the Condensation Algorithm, to track a missile in a dense countermeasures engagement. During a missile engagement, countermeasures may produce features that are identical to those of the target. Consequently, correct identification of the target missile from decoys can result in confusion, delays, and lost track. Therefore, the two algorithms were compared on the following bases: (1) the accuracy of identification, (2) the time required to identify the target correctly, and (3) sensitivity with respect to missile and decoy masses and process noise.

Table Contents

1. INTRODUCTION

In many target tracking applications, one is confronted with the problem of distinguishing the true target from decoys. The purpose of this work was to compare the performance of two tracking technologies, the Kalman filter and a factored sampling procedure known as the Condensation Algorithm, to track a missile in a dense countermeasures engagement. The two algorithms were compared on several bases such as: (1) the accuracy of identification, (2) the time required to identify the target correctly, and (3) sensitivity with respect to missile and decoy masses and process noise.

During a missile engagement, "clutter" is introduced with deployment of countermeasures that can mimic the missile. The deployment of decoys can cause the *a posteriori* target state probability density to be multi-modal. Traditional tracking techniques (variations of the Kalman filter [Cardillo, Mrstik, and Plambeck, 1999]) are based on Gaussian densities and can not represent simultaneous alternative hypotheses. On the other hand, the Condensation Algorithm [Isard and Blake, 1996] can deal with the multi-modal case by estimating the *a posteriori* state probability density using factored sampling.

The simulations constructed in this work include both missile warhead and decoys. Both are observed by a simulated radar device, which cannot distinguish between the two. The only way the decoys can be distinguished from the missile is upon re-entry, at which point the less massive decoys experience greater deceleration than the missile. The basic idea of this work is to compare how well the two tracking algorithms can distinguish between the decoys and the missile by using the difference in deceleration. Section 2 through 4 describe the general theory of motion and measurement models, the Kalman filter tracker, and the Condensation Algorithm tracker; Section 5 describes the specific motion and measurement models used in this work; Section 6 contains the results, which are summarized in Section 7. The appendix contains an overview of the software implementation.

2. MOTION & MEASUREMENT MODELS

The two tracking algorithms described below make the same assumptions about the motion of a single target and about target measurements. A target has a state $X(t)$ at time t: $X(t)$ is a vector-valued random variable whose components may include directly observable quantities and other quantities which, although not directly observable, influence future states. For a deterministic system $X(t+\Delta t)$ is given by a function of $X(t)$, but in tracking applications this is assumed to be modified by process noise. The motion model specifies the ideal motion and the process noise. Similarly, an observation $Z(t)$ at time t (if an observation is made at that time) is assumed to have an ideal value, which is modified by measurement noise: the distribution of $Z(t)$ given $X(t)$ is specified by the measurement model. Observations are made at times $t_1, t_2, \dots, t_k, \dots$, and the algorithms estimate the states $x_1 = X(t_1), \dots, x_k = X(t_k), \dots$ from the observations $z_1 = Z(t_1), \dots, z_k = Z(t_k), \dots$

The motion model consists of a function f and positive definite matrices Q_1, Q_2, $\dots, Q_{k-1}, \dots$. The assumption is that

$$x_k = f(x_{k-1}, w_k, t_{k-1}, t_k)$$

where the process noise w_k is a Gaussian random vector with mean 0 and covariance matrix Q_{k-1}.

The measurement model consists of a function h and positive definite matrices $R_1, R_2, \dots, R_k, \dots$. The assumption is that

$$z_k = h(x_k, v_k, t_k)$$

where the measurement noise v_k is a Gaussian random vector with mean 0 and covariance matrix R_k. It is also assumed that all the noise vectors are statistically independent of each other and of the initial state.

3. KALMAN FILTER ALGORITHM

The Kalman filter algorithm is essentially a set of recursive equations that implement a predictor-corrector estimator. The Kalman filter estimates a process at some time and then obtains feedbacks from a noisy measurement. Therefore, there are two groups of Kalman filter equations: time update equations (predictor) and measurement update (corrector) equations. The time update equations project forward the current state and error covariance estimates to obtain an *a priori* estimate for the next time step. The measurement update equations incorporate a new measurement into the *a priori* estimates to obtain an improved *a posteriori* estimate. The time and measurement update equations are presented in Tables 1 and 2 respectively.

Table 1: Kalman filter time update equations.

$$\hat{x}_k^- = f(\hat{x}_{k-1}, 0, t_{k-1}, t_k) \quad (1)$$

$$p_k^- = A_k p_{k-1} A_k^T + W_k Q_{k-1} W_k^T \quad (2)$$

Table 2: Kalman filter measurement update equations.

$$K_k = p_k^- H_k^T (H_k p_k^- H_k^T + V_k R_k V_k^T)^{-1} \quad (3)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0, t)) \quad (4)$$

$$p_k = (I - K_k H_k) p_k^- \quad (5)$$

In the equations above

- x_k, z_k, t_k, f, h, Q_{k-1}, and R_k are as described in Section 2.
- $\hat{x}_k^-$ and $\hat{x}_k$ are *a priori* and *a posteriori* estimates of the state x_k.

- p_k and p_k^- are the covariance estimates associated with $\hat{x}_k^-$ and $\hat{x}_k$
- A_k is the Jacobian matrix of partial derivatives of f with respect to x, that is

$$(A_k)_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, 0, t) .$$

- W is the Jacobian matrix of partial derivatives of f with respect to w

$$(W_k)_{[i,j]} = \left[\frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{x}_{k-1}, w, t_{k-1}, t_k) \right]_{w=0} .$$

- H is the Jacobian matrix of partial derivatives of h with respect to x

$$(H_k)_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, 0, t_k) .$$

- K_k (computed by (3)) is the gain matrix.
- I is the identity matrix (of appropriate dimension).
- V is the Jacobian matrix of partial derivatives of h with respect to v

$$(V_k)_{[i,j]} = \left[\frac{\partial h_{[i]}}{\partial v_{[j]}}(\hat{x}_k, v, t_k) \right]_{v=0} .$$

There is some difficulty initializing the Kalman filter because one has no *a priori* information. The most rigorous approach is to initialize the filter with the first measurement, but this is inconvenient because the variance may be infinite in some directions (for example, if a measurement corresponds to position then the velocity variance after the first measurement is infinite). A more common approach is to initialize the filter with a zero state vector and large covariance.

4. CONDENSATION ALGORITHM

The condensation algorithm, like the Kalman filter, is concerned with target state distributions at various times, conditioned on observations. Unlike the Kalman filter, the condensation algorithm is non-parametric: instead of estimating mean, variance, or other parameters, the condensation algorithm gives a sample of the distribution by using a Bayesian technique called factored sampling. This difference allows the condensation algorithm to handle "arbitrary" process and measurement models as long as the conditional independence assumptions hold. The condensation algorithm uses state & measurement models of the form described in Section 2. The algorithm is as follows:

1. Initialize the tracker by creating a set of initial particles (guesses as to the location of missile). These particles can be uniformly distributed, or distributed according to some *a priori* distribution.
2. Take a measurement and compare the measurement to the predicted measurement from each particle in the set. Use the results to assign weights to the particles, taking into account noise in the measurement model.
3. Create a new sample of particles from the existing set. The particles are selected with a probability depending on their weight. Particles with small weights tend to be thrown away.
4. Use the motion model to predict where the object (missile) will be and update the particles.
5. Apply noise to allow for uncertainties in the state model.
6. Estimate the state of the tracked object (missile)

7. Go to 2.

The tracking performance based on factored sample can be summarized by the following operations on the particle set:

$$P(X_{t-1} \mid Z_1, Z_2, \ldots, Z_{t-1}) * P(X_t \mid X_{t-1}) \rightarrow P(X_t \mid Z_1, Z_2, \ldots, Z_{t-1}) \times P(Z_t \mid X_t) \rightarrow$$

$$P(X_t \mid Z_1, Z_2, \ldots, Z_t).$$

5. BALLISTIC MISSILE AND RADAR SIMULATION

The motion model for the ballistic missile warhead and decoys is a two dimensional orbital and atmospheric re-entry model (without the launch phase), including gravitational force and aerodynamic drag forces. Gravitational force is given by Newton's gravitational law,

$$F = -\frac{GMm}{\|r\|^3} r \,,$$

where G is the universal gravitational constant, M and m are the masses of the two objects (in this case, the earth and the missile or decoy), and r is the vector from the first object to the second. Aerodynamic drag is given by the drag equation [Ashley and Landahl, 1965]

$$F = -C\rho A\|v\|v \,,$$

where v is the velocity vector, A is the cross-sectional area of the object (missile or decoy), ρ is the density of the air, and C is the coefficient of drag. The drag coefficient is the sum of two contributing terms: C_P representing subsonic forces (pressure and vortex drag), and C_W representing supersonic forces (wave drag). (The viscosity drag, linear in speed, is ignored, since it is negligible.) C_P is constant, while C_W varies as a function of speed; the model used for C_W is the piecewise linear approximation

$$C_W = C_{W0} \begin{cases} 0 & : \quad \|v\| < 0.8v_0 \\ 5\left(\frac{\|v\|}{v_0} - 0.8\right) & : \quad 0.8v_0 \le \|v\| \le v_0 \\ 1 & : \quad \|v\| > v_0 \end{cases} \,,$$

where v_0 is the speed of sound. The speed of sound varies with the square root of absolute air temperature; a smooth approximation to the ISA (international standard atmosphere) model is used for the temperature.

The measurement model is based on a single radar which has range and bearing as well as some Doppler capability; the measurement provided by this device is of the form

(Range, Range Rate, Bearing)

with Gaussian errors. The measurement device does not distinguish between the warhead and the decoys.

6. RESULTS

In order to reduce the run time of the program, sensitivity analysis was done to see if some of the parameters could be modified and not significantly impact on the results. Once these modifications were made, the performances of the two algorithms were compared by varying two parameters: the mass of decoys relative to the missile and process noise.

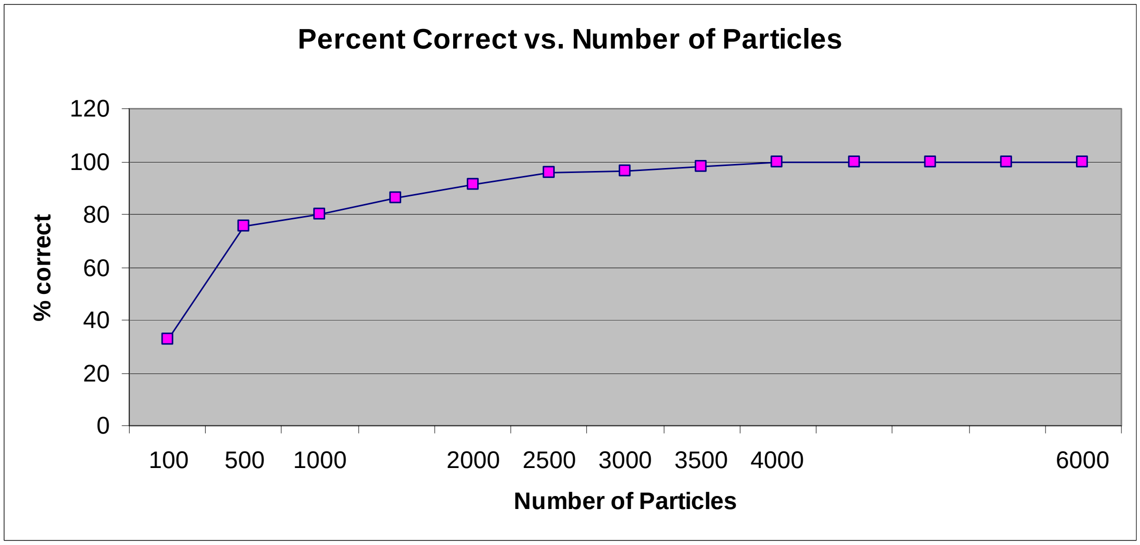
6.1 Parameter sensitivity

Before comparing the two methods of tracking, it was necessary to do sensitivity analysis on the more critical parameters. The parameters tested were:

- Number of particles used in the Condensation algorithm.
- θ_{Weight} is the threshold of particle weights about an object in the Condensation algorithm.
- Minimum number of consecutive points one target must have in order to be declared the true missile (Kalman and Condensation).

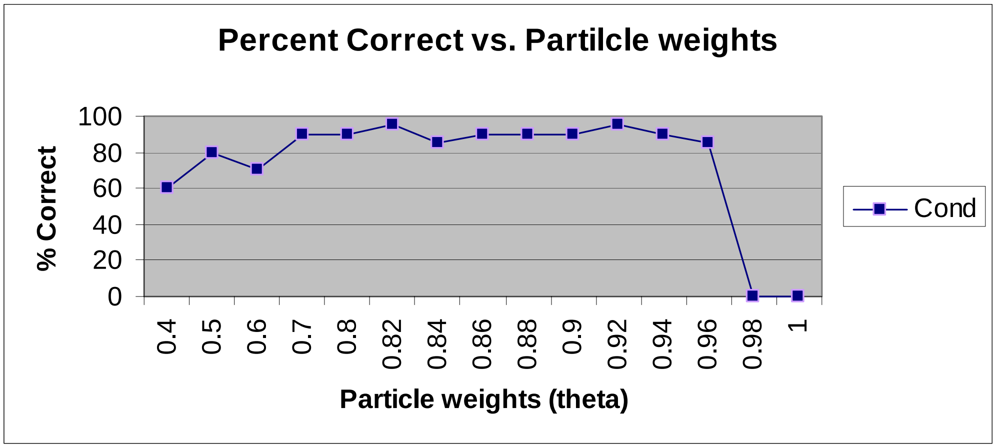
The number of particles was varied from 100 to 6,000. Figure 1 shows the results of these runs. As the number of particles increases, the probability of successfully finding the missile also increases. This is a critical parameter affecting the program's run time, so using as small a number as possible was important. Based on these runs, 3,500 particles were used for the remaining runs.

Figure 1



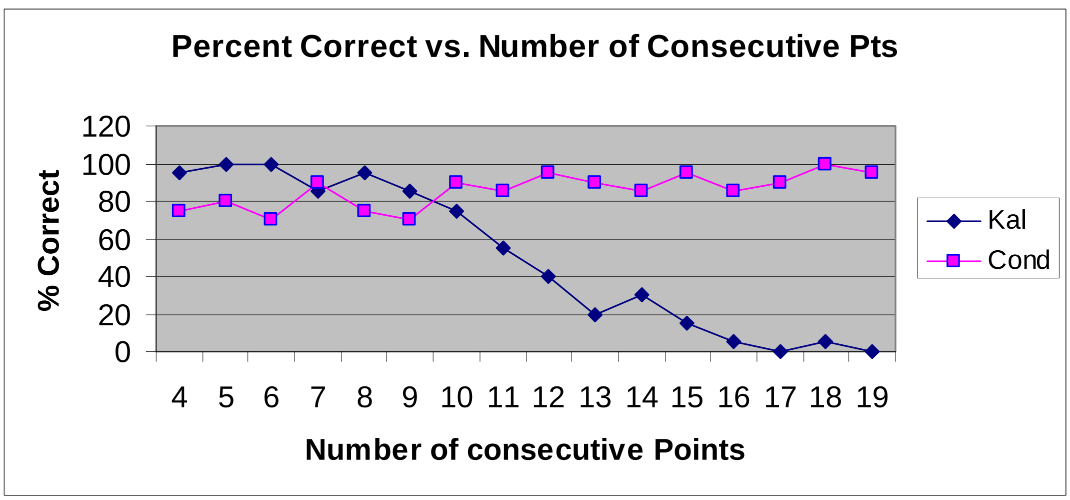
It was also necessary to determine the optimal θ_{Weight} before the decision is made that it is the missile. Figure 2 shows this θ_{Weight} against the probability of successfully finding the missile. The Condensation algorithm was insensitive to this parameter up to 96% and dropped drastically after. With each iteration, the algorithm drops particles and adds particles to all potential targets, therefore, it is impossible to get all the particles on one target and the model will not be able to make a decision when the θ_{Weight} is too high. Based on these results, it was decided that 70% was a reasonable choice for θ_{Weight}.

Figure 2



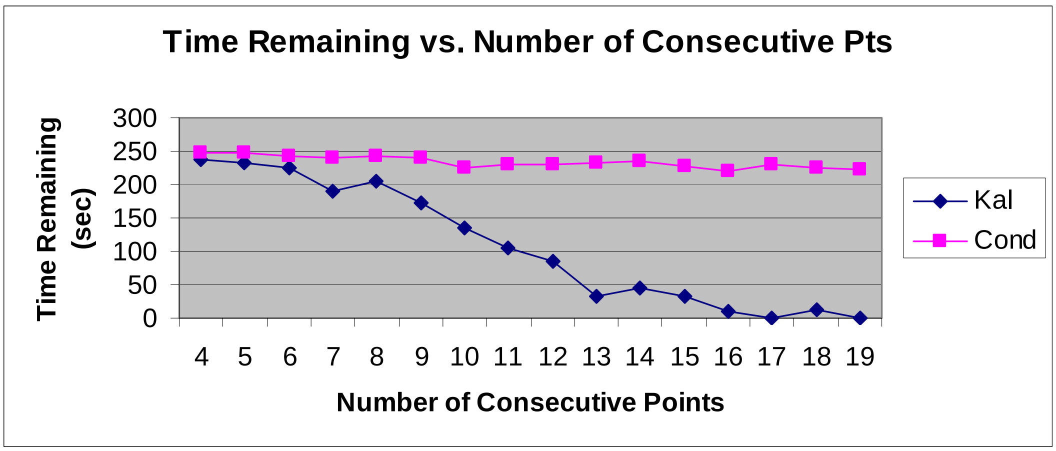
How many consecutive points (N_{Kalman} and $N_{Condensation}$, see page 21-22) must an object have in order to be declared the missile? Figures 3a and 3b show the result of varying this parameter. In figure 3a, the Condensation algorithm got 70% to 100% correct regardless of the number of consecutive points, but Kalman began to drop significantly after 10 consecutive points.

Figure 3a



In figure 3b, the time remaining (sec) represents the time remaining before the missile impacts the ground. The remaining time for the Condensation algorithm stays relatively constant, but the Kalman filter took longer to make a decision as the number of consecutive points increased. Based on these results, it was decided to use five consecutive points for remaining runs.

Figure 3b



6.2 Comparison of Kalman Filter and Condensation algorithm

Two measurements were used to determine which algorithm was better at identifying the missile. One was the percentage of correct decisions, and the other was the remaining time before ground impact.

These measurements were obtained by varying the mass of the decoys and the process noise of both the decoys and the missile.

6.2.1 Mass variation

The mass of the decoys was varied from 1% to 100% of the missile. As the mass of the decoy got closer to the mass of the missile, both algorithms had more difficulty identifying the missile. Since there are 5 decoys and one missile the probability of just randomly selecting the missile is 16.6%. Both algorithms approached this percentage as the mass of the decoys approached the mass of the missile. Figure 4a shows the probability of finding the missile for each algorithm as the mass of the decoy increases.

Figure 4a

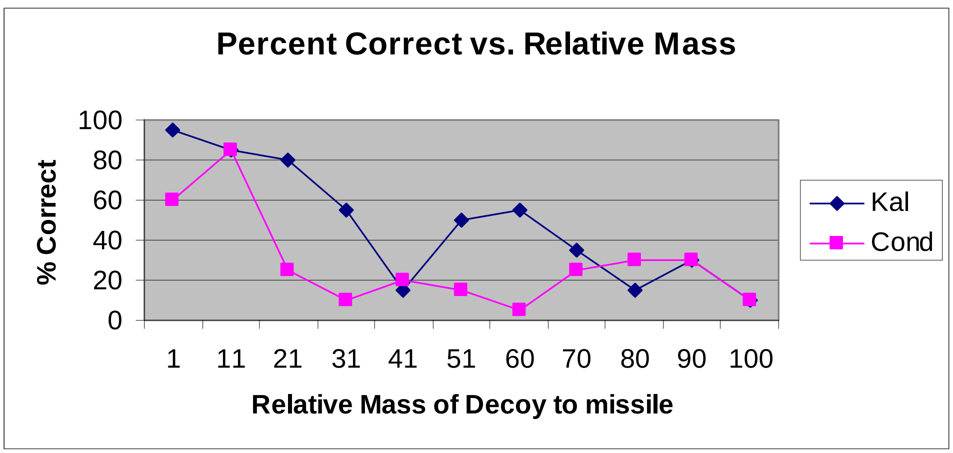
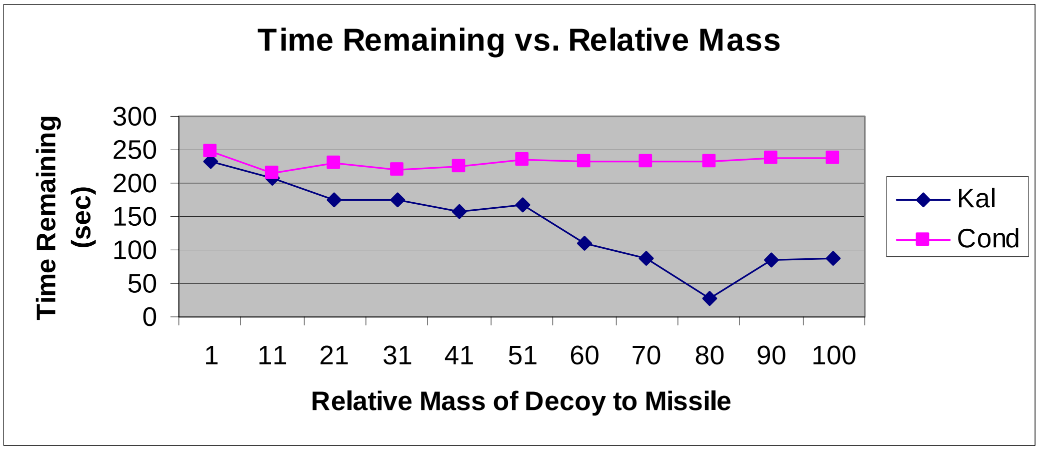


Figure 4b shows the time remaining after each algorithm makes a decision (i.e. the time in seconds from when the decision is made until the missile would impact the ground) versus the relative mass of the decoys. Although the condensation algorithm consistently makes a decision within 200 to 250 seconds to impact, Figure 4a shows it misidentifies the missile more often as the mass of the decoys increase. The Kalman algorithm takes longer to identify the missile (less time remaining) but figure 4a shows that it does no better at correctly identifying the missile than the Condensation algorithm.

Figure 4b



6.2.2 Process Noise

Increasing the process noise will make it more difficult to identify the missile since the trajectory is more random. Figures 5a and 5b show the results of varying this parameter. Figure 5a shows that the Kalman does a better job of successfully identifying the missile, however, figure 5b shows that it takes longer to make that decision.

Figure 5a

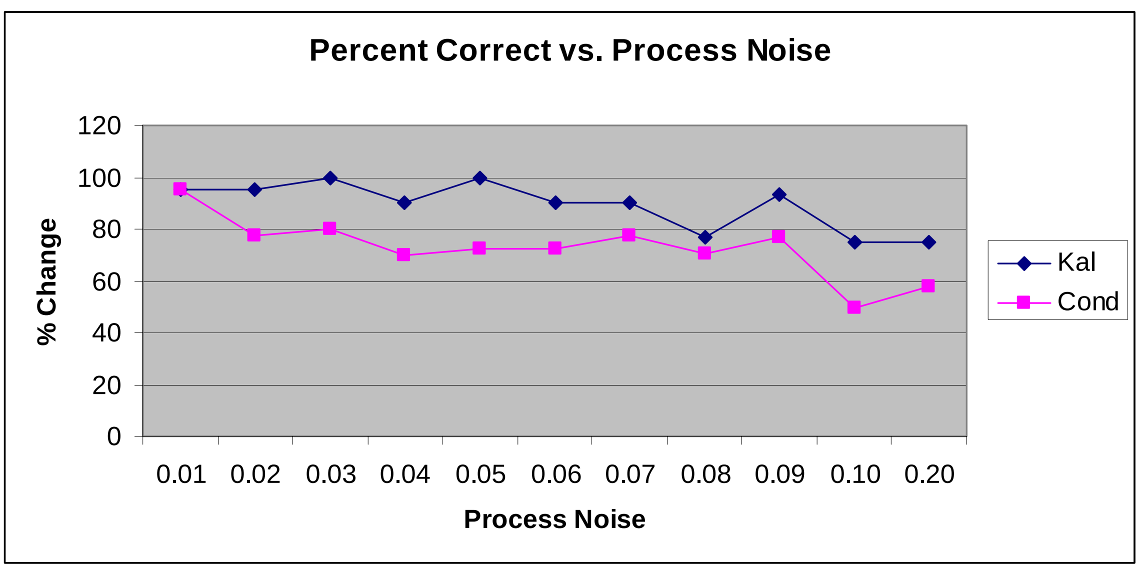
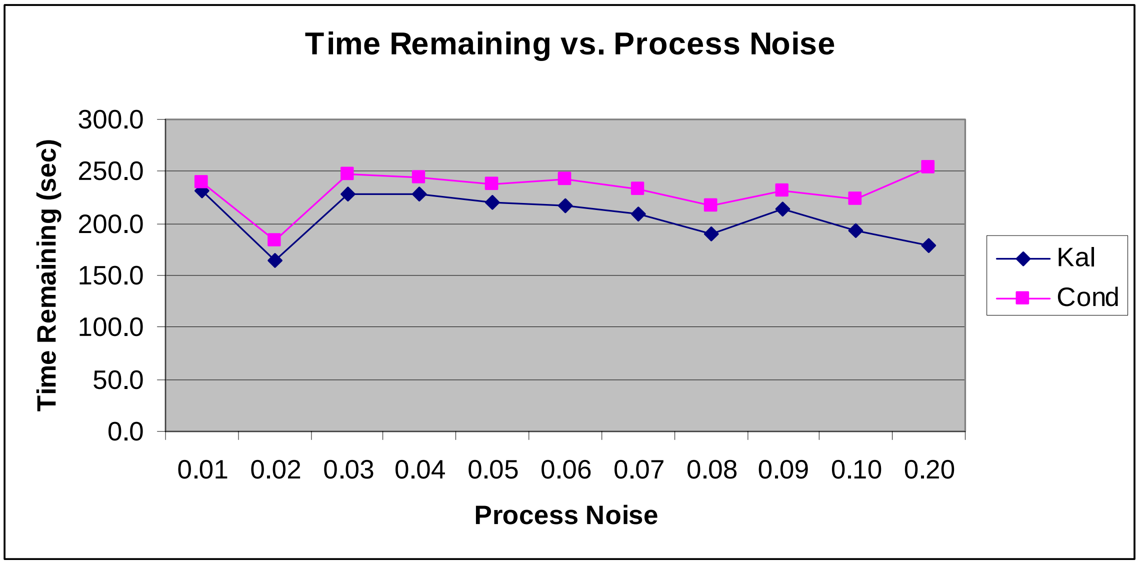


Figure 5b



Time Remaining vs. Process Noise
Time Remaining (sec)
300.0
250.0
200.0
150.0
100.0
50.0
0.0
0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10 0.20
Process Noise
Kal
Cond

7. CONCLUSIONS

- The Kalman percent correct decreases as the number of consecutive points increases because the algorithm is not able to make a decision before ground impact.
- The Condensation percent correct is not sensitive to the number of consecutive points in the range tested.
- As the mass of decoys approaches the mass of the missile, the performance of both algorithms deteriorates.
- As the mass of decoys approaches the mass of the missile, Kalman takes longer to make a decision.
- The Kalman percent correct is less sensitive to process noise.
- Kalman is more successful in identifying the missile but takes longer to make a decision.

8. RECOMMENDATIONS

- Recode in C++ to improve runtime performance.
- Perform Respond Surface Analysis to obtain the optimal criteria for different parameters.
- Test on a more powerful platform to improve runtime performance.
- Consider other reentry models such as three dimensional and parabolic.
- Do sensitivity analysis with new modifications
- Look at the possibility of combining the two algorithms to find the optimal performance (combining the speed of the Condensation algorithm with the accuracy of the Kalman algorithm).

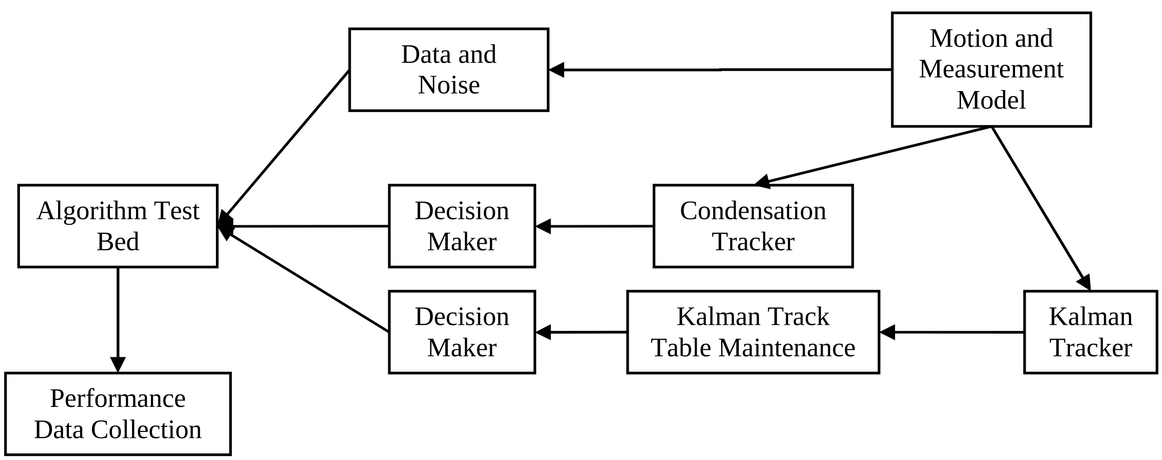
APPENDIX

SOFTWARE IMPLEMENTATION

The program is implemented as m-files run by MATLAB (version 6.5). These files are expected to run on newer versions of MATLAB. With the addition of two simple functions (true.m and false.m) they also run on MATLAB 6.1.

The design of the software is illustrated in Figure 6. Arrows indicate direction of data flow (function output).

Figure 6



The data generator and both tracking algorithms can be exercised by running the test bed (which has two versions, one that plots the targets and tracks in real-time, and one that does not, for efficiency). The test bed returns information about the decisions made by the two algorithms and returns the time remaining until warhead ground impact. The results given in Section 6 are obtained from a script which runs the test bed several times for each combination of parameter values.

The test bed initializes the targets in a linear configuration with equal spacing, and one of them is picked at random to be the missile warhead (the others being decoys). At each time step the test-bed calls the state projection function with the true motion models (missile & decoy) to advance the target positions. It also calls the measurement function to obtain radar measurements from the targets. At the beginning of a scenario (with typical parameter settings) there are no measurements because the targets are below the horizon (relative to the radar). When there are observations the test bed provides exactly the same radar data to both the Kalman filter track function and the condensation track function. (It is important that both track functions operate on identical data, to prevent anomalous performance measurements due to statistical outliers.) The track functions provide their track data to the test bed and to decision-making functions, which provide the test bed with classification results (missile, decoy, or unknown). The test bed isolates the tracking and decision-making functions from the truth data (the true target locations and the identity of the warhead). Both tracking algorithms use the missile's motion model for all tracking.

Other than the parameters varied as described in Section 6, the parameter values used in the motion and measurement simulation are the following:

- Nominal initial position of missile and decoys: 224 km altitude
- Initial velocity of missile and decoys: 7680 m/s tangential, zero radial
- Mass of missile warhead: 100 kg
- Subsonic coefficient of drag (C_P): 0.1
- Maximum supersonic coefficient of drag (C_{W0}): 0.1
- Cross-sectional area of missile warhead and decoys: 0.5 m^2

- Radar location (fixed): altitude 168.19 km, 89.938 degrees ahead of missile initial position
- Radar standard deviation of bearing measurement: 0.25 deg
- Radar standard deviation of range rate measurement: 1000 m/s
- Radar standard deviation of range measurement dependent on range:
    - Range < 1000 m : 3.16 m
    - Range between 1000 m and 10000 m : 7.07 m
    - Range > 10000 m : 30 m

The Kalman tracker decision-making function considers the match between the predicted measurement for a tracker and the closest true measurement. (If a target is moving according to the motion model assumed by the tracker – the missile – then the match will tend to be better than if the target is following a different motion model – the decoy.) The tracker with the best match (determined by the density function of the measurement conditioned on the predicted state) on one update scores a point if the density function value (scaled by $(2\pi)^{3/2}$) exceeds the threshold θ_{Dens}, which has default value $.00005 m^{-2} s^{-1} deg^{-1}$. When a tracker scores enough consecutive points the algorithm makes a decision that that tracker corresponds to the missile. The number of consecutive points required is the parameter N_{Kalman}.

The condensation algorithm decision-making function considers the total probabilistic weight of particles corresponding to one tracker. (If a target is moving according to the motion model assumed by the tracker – the missile – then the corresponding particles will tend to match observations better than they would otherwise. Their weights will therefore tend to increase as described in Section 4.) The tracker with the greatest sum of weights on one update scores a point if the sum exceeds the threshold θ_{Weight}, which has default value .7. When a tracker scores enough consecutive points the algorithm makes a decision that that tracker corresponds to the missile. The number of consecutive points required is the parameter $N_{Condensation}$.

REFERENCES

Ashley H, Landahl M., (1965) Aerodynamics of Wings and Bodies, Addison-Wesley Publ. Company, Reading, Mass.

Cardillo, G., Mrstik, A., Plambeck, T. (1999) A track filter for reentry objects with uncertain drag. IEEE Transactions on Aerospace and Electronic Systems, Vol.35, No. 2, 394-408.

Isard, M. and Blake, A. (1996) Visual tracking by stochastic propagations of conditional density. In Proc. 4th European Conf. on Computer Vision, 343-356, Cambridge, England.